

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Theoretical Computer Science 356 (2006) 190–199

Theoretical
Computer Sciencewww.elsevier.com/locate/tcs

On partially blind multihead finite automata[☆]

Oscar H. Ibarra^{a,*}, Bala Ravikumar^b^a*Department of Computer Science, University of California, Santa Barbara, CA 93106, USA*^b*Department of Computer Science, Sonoma State University, Rohnert Park, CA 94928, USA*

Abstract

This work is concerned with 1-way multihead finite automata (FA), both deterministic and nondeterministic, in which the symbol under only one head controls its move. We call such a FA a partially blind multihead FA. We show some results regarding the decision problems and closure properties of blind multihead DFA and NFA. We also compare these devices with 1-way NFA augmented by reversal bounded counters. Finally, we also present some results regarding the simulation of a partially blind DFA and NFA by a probabilistic finite automaton.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Blind head; Finite automaton; Probabilistic automaton; Counter machine; Semilinear language

1. Introduction

Various extensions of the finite automaton model have been the subject of numerous study over the past several decades. Addition of counters, pebbles and multiple heads lead to different computational models and there are many natural questions regarding the relative powers of these artifacts. The computational power also varies significantly depending on whether the head(s) is (are) allowed to move bidirectionally. In this paper, we are mainly concerned with 1-way multihead nondeterministic automata in which all the heads except one of them are incapable of distinguishing the input symbols. (They can only distinguish the input symbol from the end-marker.) The head that can distinguish the input symbols (as well as the end-marker) will be called the *sensing* head; the other heads will be referred to as the *blind* heads. This model will be referred to as a partially blind multihead finite automaton. There are some interesting similarities and differences between reversal-bounded (1-way) counter machines and partially blind multihead finite automata (FA). As far as we know, partially blind multihead FA has not been studied before. It should be noted that there is a resurgence of interest in automata models because of their applications in diverse areas such as software testing and transaction processing [1,4].

We begin by summarizing some of the previous work on the related models of multihead automata (in which all the heads can read the inputs) and multicounter machines. Multihead automata were introduced by Harrison and Ibarra [9] and have been widely studied by several researchers e.g. [18]. There has also been numerous work on reversal-bounded counter machines following their introduction by Baker and Book [2,12,7] etc. Most of the problems separating the

[☆] The research of Oscar Ibarra was supported in part by NSF Grants CCR-0208595 and CCF-0430945.

* Corresponding author. Tel.: +1 805 893 4171; fax: +1 805 893 8553.

E-mail addresses: ibarra@cs.ucsb.edu (O.H. Ibarra), ravi.kumar@sonoma.edu (B. Ravikumar).

power of 1-way counter machines and 2-way counter machines, as well as nondeterministic ones from deterministic ones were established in these and other related papers. Regarding the decision problems, it was shown in [12] that emptiness problem is decidable for 1-way NFA augmented with reversal bounded counters. It was also shown in [12] that the universe problem, the containment and the equivalence problems are not decidable for the same family. Regarding 1-way multihead FA (which is the main focus of this paper), the following are well known: when the number of heads is 1, the class of languages accepted is regular, and hence all the problems listed above are decidable, and further, the deterministic and the non-deterministic models are equivalent [10]. If the heads are all sensing, then even with $k = 2$, it is obvious that emptiness and all the above decision problems are undecidable. It is also known that there are languages accepted by a NFA with two heads, that cannot be accepted by a DFA with k heads for any k . (For example, string matching is such a problem.) The main goal of this paper is to show a contrast between the computational powers of the nondeterministic and the deterministic versions of partially blind multihead FA. We show that the equivalence and the containment problems are undecidable for the former model while they are decidable for the latter class. Further, we show how to simulate a counter machine by a partially blind FA and vice-versa and show a partial result on the tightness of the simulation (in terms of the number of counters and heads needed). We also present some results regarding the simulation of a partially blind k -DFA and k -NFA by a 2-way probabilistic finite automaton (2-pfa) and show that the former can be simulated by a 2-pfa for any k , while the latter cannot be simulated by a 2-pfa even for $k = 2$. Finally, we address the question of whether a partially blind multihead device can accept a language that is complete for $NSPACE(\log n)$.

A word of caution: we use two basic terms in this paper with a slightly different meaning than has been used in the prior work: Greibach [8] used the term *partially blind counter* to denote a counter that can assume positive and negative values, but cannot distinguish between them. Also the term sensing head has been used to refer to the ability of one head to sense the presence of another head in the same tape square.

2. Preliminaries

In this section, we introduce the basic computational models studied in this paper. We will use the following notation. For $k \geq 1$:

- (1) k -NFA—nondeterministic finite automaton with k one-way read-only input heads
- (2) k -DFA—deterministic finite automaton with k one-way read-only input heads

Let M be a k -NFA where only one head (*the read head*) can differentiate the input symbols. The other $k - 1$ heads (*the blind heads*) cannot differentiate the input symbols, i.e., all symbols are treated the same way by these heads. We assume that acceptance is by accepting state with all heads falling off the tape. If a blind head is moved beyond position $n + 1$, the input is rejected.

Formally, these models are defined as follows: a partially blind k -DFA M without a right end-marker is defined as $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ where Q is a finite set of states, $q_0 \in Q$ is the start state, $F \subset Q$ is the set of final states, and $\delta: Q \times \Sigma \rightarrow Q \times D_1 \times \cdots \times D_k$ where $D_i = \{0, 1\}$. Informally, this move means the following: if the current state is q_i , and if the sensing head is reading symbol a , and if $\delta(q_i, a) = (q_j, d_1, \dots, d_k)$, then the next state is q_j , and the heads are moved by d_i positions to the right. (We assume that head 1 is the sensing head; the rest of them are blind.) A configuration of M is defined as the $(k + 2)$ -tuple $(p_1, p_2, \dots, p_k, q, w)$ where w is the string to the right of (including the current symbol under) the sensing head, p_j refers to the position of the j -th head from the left-end of the input tape, and q denotes the current state. k -NFA is one in which δ is defined as: $\delta: Q \times \Sigma \rightarrow 2^{Q \times D_1 \times \cdots \times D_k}$.

We can define the *yields* relation as usual: if ID_1 and ID_2 are two configurations, $ID_1 \rightarrow ID_2$ means that ID_2 follows from ID_1 in a single step. We say that an input w is accepted if $(1, 1, \dots, 1, q_0, w) \rightarrow \cdots \rightarrow (n + 1, n + 1, \dots, n + 1, f, w)$, where $n = |w|$, q_0 is the start state, and f is an accepting state. The language accepted by M is denoted by $L(M)$. We call M a *partially blind k -NFA*. The deterministic version is called a *partially blind k -DFA*. A *partially blind multihead-NFA (DFA)* is a partially blind k -NFA (DFA) for some $k \geq 1$.

In the above definition, the acceptance of a string is based on all the heads falling off the tape. In Section 4, we will also consider the case in which acceptance is based on one of heads falling off the tape. In Section 4, we will also consider a k -NFA model in which there is a right end-marker. The formal definitions of these models are straight-forward extensions of the above definitions.

When $k = 1$, the automaton is an NFA (DFA) and accepts only a regular set.

3. Partially blind 2-NFAs

Partially blind 2-NFAs are one of the simplest extensions of a finite automaton, yet they can accept fairly complex languages, as the examples below show.

Example 1. Let $L = \{a^n b^n \mid n \geq 1\}$. This language can be accepted by a partially blind 2-DFA which, on a given input, first moves its read head to the first b (after verifying that all the symbols scanned prior to this b are a 's and then moves the blind head two places to the right for every b that the read head sees in the input).

The above example can be extended to show that $\{a^n b^n c^n \mid n \geq 1\}$ can be accepted by a partially blind 3-DFA. It thus follows that there are non-context-free languages accepted by k -DFA for $k > 2$.

The next example deals with a language such that a modified version of it is known to be hardest for a complexity class between L and NL [3].

Example 2. Let $L = \{a^t \# a^{x_1} \# a^{x_2} \# \dots \# a^{x_n} \mid n \geq 1, x_i \geq 1, \text{ there exist } i_1, i_2, \dots, i_k \text{ such that } t = x_{i_1} + \dots + x_{i_k}\}$. A partially blind 2-NFA accepts it as follows: on input $a^t \# a^{x_1} \# a^{x_2} \# \dots \# a^{x_n}$, M moves its read head past the first $\#$ symbol and moves its blind head once (to count the $\#$ symbol that the read head has just crossed). From now on, the read head decides (nondeterministically) to either include a block a^{x_j} (in making up the total t) or skip it. On the blocks that M decides to include, it moves the blind head twice for every one move of the read head. On the skipped blocks, as the read head moves right, the blind head is advanced just once. Also, for every $\#$ symbol read by the read head, the blind head is moved once. When the read head reaches the right end, it enters an accepting state. It can be checked that the input string is accepted if and only if it is in L as seen below: initially, the read head has made $t + 1$ moves to the right while the blind head is moved once. Thus, there is a difference of t between them. This difference is compensated by moving the blind head twice on the included blocks. The difference will become 0 if and only if there is a subset of blocks such that the total number of a 's in the blocks adds to t . This is precisely the condition for accepting the string.

Example 3. Let $L = \{xx \mid x \in \{0, 1\}^*\}$. It is well-known that $L' = \bar{L}$ is context-free. We will show that L' can be accepted by a partially blind 2-NFA M . Let $x \in L'$. It is obvious that $|x|$ is odd, or $x = x_1 a x_2 b x_3$ where $a \neq b$ (for symbols a and b) and $|x_1| + |x_3| = |x_2|$. If M guesses that $|x|$ is odd, it verifies this using its finite control. If x is of the form $x_1 a x_2 b x_3$, M verifies it as follows: it moves its read head until it reaches a (skipping over x_1) and remembers a in its finite control. From now, for every single move of the read head, the blind head is moved twice until, M guesses (nondeterministically) that the two heads are scanning the tape square. At this point, it moves its read head while keeping the blind head stationary until it reads a b . From now on, as the read head is moved once to the right, the blind head is advanced twice until the read head falls off the rightmost symbol in an accepting state. It is clear that M accepts a string if and only if it is in L' .

Our first result is rather surprising since in the partially blind 2-NFA, only the read head can distinguish the symbols; the blind head treats each symbol the same way (like all of them are the same symbol 1).

Theorem 1. *It is undecidable to determine, given a partially blind 2-NFA M over binary alphabet $\Sigma = \{0, 1\}$, whether $L(M) = \Sigma^*$. Hence, the containment and equivalence problems for languages accepted by partially blind 2-NFAs are undecidable.*

Proof. Let Z be a single-tape Turing machine starting on blank tape. Define the language $L_Z = \{x \mid x = C_1 \# \dots \# C_k, k \geq 1, \text{ each } C_i \text{ is a configuration of } Z, x \text{ is not a halting computation of } M \text{ on blank tape}\}$. We can effectively construct a partially blind 2-NFA M to accept L_Z . On input $x = C_1 \# \dots \# C_k$, M guesses a j such that C_j does not yield C_{j+1} . It verifies that C_j does not yield C_{j+1} as follows:

If $j = 1$, M moves the heads simultaneously to the right. Since the correct C_1 has only two symbols and the correct C_2 has at most 3 symbols, the read head can determine if the prefix $C_1 \# C_2$ is not valid, and if so M accepts when both heads reach the right end of the input.

If $j > 1$, then M proceeds as follows: M moves its read head to the beginning of C_j and the blind head to the beginning of C_{j+1} . This is done as follows: both heads advance at the same rate before C_{j-1} and then starting at the

beginning of C_{j-1} , the blind head moves two places to the right for every right move of the read head until the read head is at the beginning of C_j .

M then guesses a position k (from the left) in C_j such that there is a mismatch between C_j and C_{j+1} in position k . It moves both heads simultaneously k steps. M remembers the k th symbol a of C_j . At this point, the blind head is in the k th position of C_{j+1} . Now, M moves the read head until it nondeterministically reaches the same symbol that the blind head is scanning. The read head then reads the symbol in this position, and checks that it is not the k th symbol in C_j . Then M simultaneously moves both heads until they reach the right end at the same time. (Of course, some additional details are required when the position of discrepancy involves the state symbol or a tape symbol that was modified.)

Now let $R_Z = \{x \mid x \neq C_1\# \cdots \# C_k, \text{ where each } C_j \text{ is configuration of } Z\}$. Clearly R_Z is a regular set. Hence $S_L = L_Z \cup R_Z$ can be accepted by a partially blind 2-NFA. Let Σ be set of symbols used in the language S_L . Then $S_L = \Sigma^*$ if and only if Z does not halt on blank tape. Let $\text{Binary}(S_Z)$ be a binary encoding of the language S_Z ; thus $\text{Binary}(S_Z) \subseteq \{0, 1\}^*$. Clearly, $\text{Binary}(L_Z)$ can also be accepted by a partially 2-NFA, and $\text{Binary}(S_Z) = \{0, 1\}^*$ if and only if Z does not halt on blank tape. \square

Our definition of acceptance requires that the machine not only enters an accepting state after *both* heads fall off the input tape. We now look at a simpler model, where we require that acceptance is by accepting state after *at least* one head falls off the input tape. We consider two cases.

Case 1: Here we look at the model where the read head is always behind the blind head, and acceptance is when the machine enters an accepting state after the blind head falls off the tape.

Example 4. Let Σ be any finite set of symbols and $\#$ be a new symbol not in Σ . Let $L_\Sigma = \{x\#y \mid x \in \Sigma^+, y \in (\Sigma \cup \{\#\})^+, |x| = i + j + k \text{ for some nonnegative integers } i, j, k, \text{ and } |x\#y| = i + 2j + 3k + 1\}$. Clearly, L_Σ is not a regular set, but can be accepted by a partially blind 2-NFA, M , which operates as follows on input $x\#y$: for every symbol different from $\#$ that the read head sees in x , the blind head moves right 1 or 2 or 3 cells, the choice being made nondeterministically. When the read head reaches $\#$, M advances the blind head one position to the right and enters an accepting state.

The following result is stronger than Theorem 1.

Theorem 2. *It is undecidable to determine, given a partially blind 2-NFA M over alphabet Σ with acceptance defined above, whether $L(M) = L_\Sigma$. Hence, containment and equivalence problems for this model is undecidable.*

Proof. For any alphabet Σ , define the relation $R_\Sigma = \{(x, 1^r) \mid x \in \Sigma^+, |x| = i + j + k \text{ for some } i, j, k, \text{ and } r = i + 2j + 3k\}$. It is known that it is undecidable to determine, given an ε -free nondeterministic GSM G (without accepting states) over input alphabet Σ and unary output alphabet $\{1\}$, whether $R(G) = R_\Sigma$ [13]. Note that R_Σ itself can be defined by a simple ε -free GSM.

Let $\#$ be a new symbol. We construct from G a partially blind 2-NFA M which, when given a string $x\#y$, where $x \in \Sigma^+, y \in (\Sigma \cup \{\#\})^+$, simulates G . The read head simulates the input head of G on x . The blind head simulates the output of G , i.e., in a step, if G outputs 1^j , then blind head advances j places to right (regardless of the symbols it encounters on the tape). Note that the computation proceeds in phases, where in each phase, the read head reads a symbol, and the blind head advances (simulates a unary output). Note also that since G is ε -free, the blind head never lags behind the read head. When the read head sees $\#$, M advances the blind head one position to the right and enters an accepting state. One can easily verify that $L(M) = L_\Sigma$ if and only if $R(G) = R_\Sigma$. The result follows. \square

Case 2: Here we look at the model where the blind head is always behind the read head, and acceptance is when the machine enters an accepting state after the read head falls off the tape. It is easy to see that, unlike Case 1, this model can only accept a regular set. The reason is that, since the blind head is always lagging the read head, the blind head does not enter into the dynamics of the computation. To accept more than regular sets, we can modify acceptance of the model by also requiring that the blind head be scanning any symbol from a specified subset S of the input alphabet. We call the set *marker set*.

Example 5. Let Σ be any finite set of symbols and $\#$ be a new symbol not in Σ . Let $L_\Sigma = \{x\#y \mid x, y \in \Sigma^+, |x| = i + j + k \text{ for some nonnegative integers } i, j, k, \text{ and } |x\#y| = 2i + 3j + 6k + 1\}$. L_Σ is not a regular set, but can be

accepted by a partially blind 2-NFA, M , which operates as follows on input $x\#y$: for every 2 or 3 or 6 right moves of the read head on x (the choice is done nondeterministically), the blind head moves right one position. When the read head sees the symbol $\#$, it continues processing y entering an accepting state after reading 2 or 3 or 6 symbols. Clearly, with marker set $S = \{\#\}$, $L(M) = L_\Sigma$.

As in Theorem 2, we have:

Theorem 3. *It is undecidable to determine, given a partially blind 2-NFA M over alphabet Σ with acceptance defined above, whether $L(M) = L_\Sigma$.*

Proof. It is known that it is undecidable to determine, given an ε -free nondeterministic GSM G (without accepting states) over a unary input $\{1\}$ and output alphabet Σ , whether $R(G) = \{(1^r, w) \mid r \geq 1, r = i + j + k, w \in \Sigma^+, |w| = 2i + 3j + 6k\}$.

Given such a GSM G , we can easily construct another GSM G' with input alphabet Σ and a unary output alphabet $\{1\}$. Suppose on input 1^n , G outputs x . Note that since G is ε -free, $|x| \geq n$. Then G' on input x simulates the computation of G as follows. Suppose G in state p reading symbol 1 outputs string u and enter state q . Then this step is simulated by G' by reading u on the input and only outputs 1 after reading the last bit of u (i.e., G' outputs ε 's for the first $|u| - 1$ bits of u). The state changes in G (i.e., p to q) are recorded in the finite-state control of G' . It follows that it is undecidable to determine, given an ε -free GSM G' , whether $R(G') = \{(w, 1^r) \mid r \geq 1, r = i + j + k, w \in \Sigma^+, |w| = 2i + 3j + 6k\}$.

The rest of the proof is similar to the one for Theorem 2 using the GSM G' . That is, we can construct a partially blind 2NFA M with marker set $\{\#\}$ such that $L(M) = L_\Sigma$ (of Example 3) if and only if $R(G') = \{(w, 1^r) \mid r \geq 1, r = i + j + k, w \in \Sigma^+, |w| = 2i + 3j + 6k\}$. The result follows. \square

4. Partially blind multihead NFAs and 1-reversal multicounter machines

For the remainder of the paper, we will assume that acceptance is by accepting state with *all* the heads falling off the endmarker.

We can augment the input with an endmarker, where now the blind heads can differentiate an input symbol from the endmarker. The following lemma shows that this model is no more powerful than the original model.

Lemma 1. *A language L can be accepted by a k -NFA if and only if it can be accepted by a k -NFA without an endmarker.*

Proof. Consider the case $k = 2$. The generalization is straightforward. Clearly, the model without endmarker can be simulated by one with endmarker. If M is a partially blind 2-NFA with endmarker, we can construct a partially-blind 2-NFA without endmarker M' equivalent to M . M' simulates M . Then at some point (nondeterministically chosen), M' executes (1) or (2) below (the choice between (1) and (2) is also nondeterministic):

(1) M' guesses that read head of M is about to process the last input symbol before the endmarker and simulates that step.

At some later time, M' guesses that the blind head of M is about to process the last input symbol before the endmarker and simulates that step as well as the steps the read head and the blind head would have made on the endmarker, all in one step.

(2) M' guesses that blind head of M is about to process the last input symbol before the endmarker and simulates that step.

At some later time, M' guesses that the read head of M is about to process the last input symbol before the endmarker and simulates that step as well as the steps the blind head and the read head would have made on the endmarker, all in one step.

It follows that $L(M') = L(M)$. \square

In view of Lemma 1, we will assume in the remainder of the paper that the input has an endmarker. Thus, a partially blind k -NFA has one read head and $k - 1$ blind heads operating on an input with an endmarker. We also make the same assumption for the deterministic case.

We will show that the Parikh map of the language accepted by a partially blind multihead-2NFA is semilinear. First we recall the definition of a semilinear set [10].

Let N be the set of nonnegative integers and n be a positive integer. A subset S of N^n is a *linear set* if there exist vectors v_0, v_1, \dots, v_t in N^n such that $S = \{v \mid v = v_0 + m_1 v_1 + \dots + m_t v_t, m_i \in N\}$. The vectors v_0 (referred to as the *constant vector*) and v_1, v_2, \dots, v_t (referred to as the *periods*) are called the *generators* of the linear set S . A set $S \subseteq N^k$ is *semilinear* if it is a finite union of linear sets. The empty set is a trivial (semi)linear set, where the set of generators is empty. Every finite subset of N^n is semilinear—it is a finite union of linear sets whose generators are constant vectors. It is also clear that semilinear sets are closed under union. It is also known that they are closed under complementation and intersection.

Let $\Sigma = \{a_1, a_2, \dots, a_n\}$ be an alphabet. For each string w in Σ^* , define the *Parikh map* of w to be $\psi(w) = (|w|_{a_1}, |w|_{a_2}, \dots, |w|_{a_n})$, where $|w|_{a_i}$ denotes the number of occurrences of a_i in w . For a language $L \subseteq \Sigma^*$, the *Parikh map* of L is $\psi(L) = \{\psi(w) \mid w \in L\}$. A language L is *semilinear* if $\psi(L)$ is a semilinear set.

Let k be a nonnegative integer. A *1-reversal k -counter machine* (denoted by k -NCM) is a nondeterministic finite automaton with a one-way read only input (with right endmarker) augmented with k counters. Each counter starts with value zero and can be incremented/decremented by 1 and tested for zero but once the counter decreases, it can no longer be incremented (thus the counter reverses mode once). An input w (plus an endmarker) is accepted by such an acceptor if, when started in its initial state with all counters zero, the machine enters an accepting state after falling off the endmarker (note that we do not require that the machine move its input head to the right at each step). The deterministic version is denoted by k -DCM.

We will need the following result in [12] (see also [11]).

Theorem 4. Let $\Sigma = \{a_1, \dots, a_n\}$.

- (1) If L is accepted by a k -NCM (for some k), then $\psi(L)$ is semilinear.
- (2) If $S \subseteq N^n$ is semilinear, then the language $L = \{a_1^{i_1} \dots a_n^{i_n} \mid (i_1, \dots, i_n) \in S\}$ can be accepted by a k -NCM (for some k).

The above result also holds for the deterministic version.

Lemma 2. If M is a k -NCM, we can construct an equivalent k -NCM M' such that for every string x of length n that is accepted, M' has an accepting computation of x in which each counter does not exceed the value n during the computation.

Proof. It was shown in [2] that there is a fixed constant c such that if x (of length n) is accepted by M , then there is an accepting computation in which M makes no more than cn steps. It follows that the value of M during such a computation does not exceed cn . We modify M to M' which uses a buffer for each counter in the finite control and increments/decrements each counter modulo c . \square

Theorem 5. Let $k \geq 1$.

- (1) Every partially blind $(k+1)$ -NFA can be simulated by a $(k+1)$ -NCM.
- (2) Every k -NCM can be simulated by a $(2k+1)$ -NFA.

Proof. For part (1), suppose M is a partially blind $(k+1)$ -NFA (thus it has k blind heads), where $k \geq 1$. We construct a $(k+1)$ -NCM M' . The input head of M will simulate the read head of M while counter i ($1 \leq i \leq k$) will simulate blind head i . Counter $(k+1)$ will have a special purpose. Let the input be w (plus the endmarker). M' starts by incrementing all the $k+1$ counters simultaneously to a nondeterministic value n . Then M' simulates M with the right moves of blind head i being simulated by a decrementing counter i ($1 \leq i \leq k$). Counter $(k+1)$ is decremented every time the input head (which simulates the read head of M) moves right. Clearly, counter $(k+1)$ becomes zero when the input head reaches the endmarker if and only if $n = |x|$. Clearly, M' can be constructed to accept $L(M)$. We omit the details.

For part (2), suppose M is a k -NCM (note that each counter is 1-reversal). Assume that M satisfies Lemma 2. We construct a partially blind $(2k+1)$ -NFA M' (i.e., M' has $2k$ blind heads) equivalent to M . For each counter C of M , M' uses two blind heads H_1 and H_2 . H_1 simulates C when it is in the increasing mode by making a right move for every

increment of C . When C changes mode to decreasing, M' then uses H_2 to simulate C , again making a right move for every decrement of C . At some point, M' guesses that H_2 is in the same input position as H_1 . It verifies this by moving H_1 and H_2 simultaneously. Clearly, the counter C being simulated is zero if the two blind heads reach the endmarker at the same time. It follows that M' can simulate M , and $L(M') = L(M)$. Again, we omit the details. \square

Corollary 1. *The Parikh map of the language accepted by any partially blind multihead-NFA is semilinear.*

The next result shows that the trade-offs in Theorem 5 are, in a sense, the best possible.

Theorem 6. *There is a language that can be accepted by a partially blind 2-NFA, but not by a 1-NCM.*

Proof. A partially blind 2-NFA can accept the language L over the alphabet $\{a, b\}$ defined as: $L = \{x \mid x \text{ has equal number of } a\text{'s and } b\text{'s}\}$. The blind head moves twice for each a , and if the heads reach the end-marker at the same time, the string is accepted. But it is well-known that L is not a meta-linear language (Exercise 6.14 in [10]), so even with any fixed (bounded) number of reversals, a (1-way) nondeterministic counter machine with one counter cannot accept L . \square

We do not know if the converse is true.

The following result shows that a 2-DFA (i.e., both heads are read heads) cannot be simulated by any partially blind multihead-NFA.

Corollary 2. *There is a language L that can be accepted by a 2-DFA that cannot be accepted by any partially blind multihead-NFA.*

Proof. Let $L = \{a^1 \# a^2 \# a^3 \# \dots \# a^{n-1} \# a^n \mid n \geq 1\}$. Clearly, L can be accepted by a 2-DFA. But, since the Parikh map of L is not semilinear, it cannot be accepted by any partially blind multihead-NFA. \square

The next result shows that nondeterminism is more powerful than determinism for partially blind multihead-FA.

Corollary 3. *There is a language L that can be accepted by a partially blind 2-NFA that cannot be accepted by any partially blind multi-head DFA.*

Proof. Let $L = \{a^1 \# a^2 \# a^3 \# \dots \# a^{n-1} \# a^n \mid n \geq 1\}$. Let L' be the complement of L . L' can be accepted by a partially blind 2-NFA. Suppose L' can be accepted by a partially blind k -DFA. Then by Theorem 8 below, L (the complement of L') can also be accepted by a partially blind k -DFA. This is not possible from Corollary 2. \square

Theorem 7. *Let C_N be the class of languages accepted by partially blind multihead-NFAs. Then*

- (1) C_N is closed under union and intersection, but not closed under complementation.
- (2) The membership, emptiness, and disjointness problems for C_N are decidable.
- (3) The containment and equivalence problems for C_N are undecidable.

Proof. Closures under union and intersection are easily verified. Nonclosure under complementation follows from Corollaries 2 and 3 (that, is L' is in C_N but $L = \text{complement of } L'$ is not in C_N).

Decidability of membership, emptiness, and disjointness follow from part (1), Corollary 1, and the fact that it is decidable to determine whether a semilinear set is empty. Undecidability of containment and equivalence follow from Theorem 1. \square

On the other hand for the deterministic version, we have:

Theorem 8. *Let C_D be the class of languages accepted by partially blind multihead-DFAs. Then*

- (1) C_D is closed under union, intersection, and complementation.
- (2) The membership, emptiness, disjointness, containment, and equivalence problems for C_D are decidable.

Proof. Again, the first part is easily verified. The second part follows from part (2) of Theorem 7 and closure under complementation. \square

Let $\text{NSPACE}(\log n)$ be the class of languages accepted by $\log n$ space nondeterministic Turing machines. Denote the deterministic class by $\text{DSpace}(\log n)$. Clearly, languages accepted by partially blind multihead-NFA are in $\text{NSPACE}(\log n)$. However, it is open whether these languages are in $\text{DSpace}(\log n)$. In fact, this is not known even for the case of partially blind 2-NFAs. For (unrestricted) 2-NFAs, it is known [18,14] that they can accept languages that are complete for $\text{NSPACE}(\log n)$. By definition, if a language L is complete for $\text{NSPACE}(\log n)$, then if L is in $\text{DSpace}(\log n)$, then $\text{DSpace}(\log n) = \text{NSPACE}(\log n)$.

Example 6. Let $L = \{x\#w_1\#\dots\#w_k \mid \text{there exist indices } i_1 < i_2 < \dots < i_r \text{ such that } x = w_{i_1}w_{i_2}\dots w_{i_r}\}$. It was shown in [14] that this language is complete for $\text{NSPACE}(\log n)$. That L can be accepted by a 2-NFA (i.e., both heads are read heads) is obvious.

An interesting question is whether there is a language that is complete for $\text{NSPACE}(\log n)$ that can be accepted by a partially blind multihead-NFA. We do not know the answer to this question at this point.

However, for the case $k = 2$, we have:

Lemma 3. *If M is a partially blind 2-NFA, then $L(M)$ can be accepted by a finite automaton M' augmented with one unrestricted counter. Hence, $L(M)$ is context-free.*

Proof. M' simulates M using its input head to simulate the read head of M and the counter to implicitly simulate the blind head. The idea is for the counter of M' to keep track of the distance between the read head and the blind head of M , and the finite control to keep track of which of the heads is trailing. When the counter is zero, the two heads are in the same position of the input. Clearly, M' can be constructed to accept $L(M)$. \square

From this, the following theorem readily follows:

Theorem 9. *There is a language L that is complete for $\text{NSPACE}(\log n)$ that cannot be accepted by a partially blind 2-NFA.*

Proof. The language presented in Example 6 can be shown to be a non-context-free language and the conclusion follows from Lemma 3. \square

5. Simulation of a partially blind DFA by a two-way probabilistic finite automaton

In this section, we compare the powers of partially-blind k-DFA and 2-pfa. We show that a partially blind k-DFA can be simulated by a 2-pfa, but a partially blind 2-NFA cannot be simulated by a 2-pfa. Probabilistic models have become central in various aspects of computation [15]. We consider a 2-pfa which has a single two-way input head. The model we use is the one in which the probability of error is required to be bounded away from $\frac{1}{2}$, or equivalently, for any $\varepsilon > 0$, the probability of error on *any* input string is at most ε . (The number of states can be a function of ε , but obviously, not the length of the input string.)

We informally describe the 2-pfa model. We say that a language L can be accepted by a 2-pfa if the following holds: for any ε , $0 < \varepsilon \leq \frac{1}{2}$, there exists a 2-way DFA $M(\varepsilon)$ (the number of states of which can be a function of ε) such that for any input $x \in L$ (x not in L), $M(\varepsilon)$ accepts x with probability at least $1 - \varepsilon$ (at most ε). This model of a pfa is called the bounded error model. Rabin [16] showed that a 1-pfa with bounded error can accept only regular languages.

Freivalds [6] gave an elegant and simple construction to show that a 2-pfa can accept a nonregular language $L_{\text{eq}} = \{0^n 1^n \mid n \geq 1\}$ with bounded error, or equivalently with arbitrarily small error probability. This result was generalized in [17] to the following: for a fixed k , a 2-pfa can accept any bounded semi-linear language $L \subseteq a_1^* a_2^* \dots a_k^*$ (where a_1, a_2, \dots, a_k are individual symbols).

We present the basic construction of Freivalds [6]. For completeness, we outline a proof of this lemma.

Lemma 4. L_{eq} can be accepted by a 2-pfa.

Proof. Given an arbitrary constant ε , $0 < \varepsilon \leq 1$, choose positive integers $c(\varepsilon)$ and $d(\varepsilon)$ large enough so that

$$2 \cdot (1/2)^{d(\varepsilon)} < \varepsilon,$$

$$\left(\frac{2^{c(\varepsilon)}}{(1 + 2^{c(\varepsilon)})} \right)^{d(\varepsilon)} > 1 - \varepsilon.$$

We now describe a 2-PFA $M(\varepsilon)$ which accepts L_{eq} with error probability less than ε . We may assume that the input is of the form $0^n 1^m$. The inputs not in this form can be readily rejected using the finite control. In the first stage, $M(\varepsilon)$ checks that $n \equiv m \pmod{c(\varepsilon)}$; if not, $M(\varepsilon)$ rejects the input. If the input is not rejected thus far, the input is of the form $0^n 1^m$ where $n = m$, or $|n - m| \geq c(\varepsilon)$. This gap between n and m in the latter will be used to distinguish it from the former.

Now, $M(\varepsilon)$ proceeds to the second stage in which $M(\varepsilon)$ conducts a sequence of “competitions” between 0’s and 1’s of its input: a competition involves tossing $c(\varepsilon)$ fair-coin tosses for each input symbol. Thus a total of $(n + m)c(\varepsilon)$ fair coins will be tossed. The outcome of a competition is positive for 0 (1) if the first $nc(\varepsilon)$ (last $mc(\varepsilon)$) tosses turn up all heads. A competition is decisive if its outcome is positive for 0 or 1, but not both. In the case of a decisive outcome, the winner is the symbol with a positive outcome. $M(\varepsilon)$ conducts a sequence of competitions until the total number of wins exceeds $d(\varepsilon)$. $M(\varepsilon)$ accepts the input if and only if both 0 and 1 have registered at least one win. It can be shown that $M(\varepsilon)$ accepts L_{eq} with error probability less than ε . \square

The next lemma is an obvious consequence of the above lemma.

Lemma 5. Let k be a fixed positive integer and let $\Sigma_k = \{i_1, i_2, \dots, i_k\}$. Let L be defined over Σ_k as the set of strings $x \in \Sigma_k^*$ such that the number of occurrences of i_r in x = the number of occurrences of i_s for all r and s . Then, L can be accepted by a 2-pfa.

Proof. The proof is closely related to that of Lemma 4. Choose integers $c(\varepsilon)$ and $d(\varepsilon)$ large enough so that

$$2 \cdot (1/2)^{k^2 d(\varepsilon)} < \varepsilon,$$

$$\left(\frac{2^{c(\varepsilon)}}{(1 + 2^{c(\varepsilon)})} \right)^{k^2 d(\varepsilon)} > 1 - \varepsilon.$$

In stage 1, a 2-pfa $M(\varepsilon)$ checks that $i_t \equiv i_s \pmod{c(\varepsilon)}$ for all t and s and rejects the input if this is not true for at least one pair. Then, in stage 2, competitions are conducted between i_r and i_s for every pair (one at a time) until the number of wins for i_t or i_s reaches $d(\varepsilon)$. The string is accepted if and only if a win is registered by both i_t and i_s for every pair (t, s) . It is easy to see that $M(\varepsilon)$ accepts L . \square

We now show the main result of this section. This result is a significant generalization of Example 1.

Theorem 10. Let L be accepted by a partially blind k -DFA for some k . Then, L can be accepted by a 2-pfa.

Proof. Let M be a partially blind k -DFA with $k - 1$ blind heads and one reading head such that $L = L(M)$. M accepts a string if an accepting state is reached when all the heads have fallen off the input tape. Equivalently, a string w is accepted if the number of times each head has moved is exactly equal to $|w| + 1$ and an accepting state is reached.

Let ε be the error bound permitted for the 2-pfa. We construct M_1 , a 2-pfa to simulate M . The finite control of M_1 simulates the finite control of M and a faithful simulation is possible since the next move does not depend on the positions of the blind heads. To simulate the k heads, k variables i_1, i_2, \dots, i_k are used. i_t is used to count the right moves performed by the t -th head. If these values are available in unary, clearly Lemma 5 can be used to check if the equations $i_t = i_s$ holds for pairs. Even though they are not explicitly available, they are implicitly available during the head movement performed by M . Further, since M is deterministic, the head movements can be implicitly generated

any number of times, as required in Lemma 5. The acceptance is conditioned by halting in an accepting state and the equalities $i_t = i_s$, for all pairs (t, s) . The parameters in the simulation of Lemma 5 can be so chosen that L can be accepted with error bound at most ε . \square

In contrast to the above theorem, we can show the following:

Theorem 11. *There is a language that can be accepted by a partially blind 2-NFA that cannot be accepted by a 2-pfa.*

Proof. The language $Center = \{x1y \mid x, y \in \{0, 1\}^*, |x| = |y|\}$ is such a candidate. It is easy to see that $Center$ can be accepted by a partially blind 2-NFA which operates as follows: the sensing head nondeterministically moves to the center and checks that the symbol scanned is 1. Then, for every move of the sensing head, the blind head is moved twice. Both heads reach the right endmarker at the same time if and only if the input string is in the language. It was shown in [5] that $Center$ cannot be accepted by a 2-pfa. \square

6. Conclusions

There is a renewed interest in the study of various extensions of finite automata since they have found applications in formal modeling and verification of software systems [4]. Here we considered a natural restriction of a multihead FA—namely, only one head can distinguish between input symbols. This model resembles a reversal-bounded counter machine, but there are some differences between them. We showed that all the fundamental decision problems (membership, emptiness, disjointness, containment, equivalence etc.) are decidable for blind multihead DFA, while only some problems (membership, emptiness and disjointness) are decidable in the nondeterministic case, while the others (finiteness, equivalence and containment) are undecidable. We also presented some closure properties of partially blind multihead FA and showed that the partially blind k -DFA can be simulated by a 2-pfa, while there are 2-NFA's that cannot be simulated by a 2-pfa.

We conclude with some open problems:

- (a) can a language complete for NSPACE($\log n$) be accepted by a blind multihead NFA?
- (b) How tight are the bounds in Theorem 5?
- (c) Is the converse of Theorem 6 true?

References

- [1] R. Alur, M. Yannakakis, Model checking of hierarchical state machines, in: Sixth ACM Symp. on the Foundations of Software Engineering, 1998, pp. 175–188.
- [2] B. Baker, R. Book, Reversal-bounded multipushdown machines, J. Comput. System Sci. 8 (1974) 315–332.
- [3] S. Cho, D. Huynh, On a complexity hierarchy between L and NL, Inform. Process. Lett. 29 (4) (1988) 177–182.
- [4] Z. Dang, O. Ibarra, S. San Pietro, G. Xie, Real-counter automata and their decision problems, in: Proc. 24th International Conference on Foundations of Software Technology and Theoret. Comput. Sci. (FSTTCS' 04), Lecture Notes in Computer Science, Vol. 3328, Springer, 2004, pp. 198–210.
- [5] C. Dwork, L. Stockmeyer, On the power of 2-way probabilistic finite state automata, in: Proc. 30th IEEE Annu. Symp. on Found. of Comput. Sci., 1989, pp. 480–485.
- [6] R. Freivalds, Probabilistic two-way machines, in: Proc. Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Vol. 118, Springer, Berlin, 1981, pp. 33–45.
- [7] S. Greibach, Remarks on the complexity of nondeterministic counter languages, Theoret. Comput. Sci. 1 (1976) 269–288.
- [8] S. Greibach, Remarks on blind and partially blind 1-way multicounter machines, Theoret. Comput. Sci. 7 (1978) 311–324.
- [9] M. Harrison, O. Ibarra, Multi-tape and multi-head pushdown automata, Inform. and Control 13 (1968) 433–470.
- [10] J. Hopcroft, J. Ullman, Introduction to Automata, Formal Languages and Theory Computation, Addison-Wesley, Reading, MA, 1979.
- [11] O. Ibarra, A note on semilinear sets and bounded-reversal multihead pushdown automata, Inform. Process. Lett. 3 (1974) 25–28.
- [12] O. Ibarra, Reversal-bounded counter machines and their decision problems, J. ACM 25 (1) (1978).
- [13] O. Ibarra, The unsolvability of the equivalence problem for epsilon-free NGSM's with unary input (output) alphabet and applications, SIAM J. Comput. 7 (4) (1978) 524–532.
- [14] B. Jenner, Knapsack problems for NL, Inform. Process. Lett. 54 (3) (1995) 169–174.
- [15] P. Pardalos, et al., Handbook on Randomized Computing, Vols. I and II, Kluwer, Dordrecht, 2001.
- [16] M.O. Rabin, Probabilistic automata, Inform. and Control 6 (3) (1963) 230–245.
- [17] B. Ravikumar, Some observations on 2-way probabilistic finite automata, in: Proc. 12th Conf. on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, Springer, Berlin, 1992, pp. 392–403.
- [18] I. Sudborough, A note on tape-bounded complexity classes and linear context-free languages 22 (4) (1975) 499–500.